PENINGKATAN AKURASI ALGORITMA *BACKPROPAGATION* DENGAN SELEKSI FITUR *PARTICLE SWARM OPTIMIZATION* DALAM PREDIKSI PELANGGAN TELEKOMUNIKASI YANG HILANG

Irvan Muzakkir[1], Abdul Syukur[2], Ika Novita Dewi[3]

[1,2,3] Pasca Sarjana Teknik Informatika, Universitas Dian Nuswantoro, Semarang 50131

[1]irvanmuzakkir1@gmail.com
[2]abdulsyukur2014@gmail.com
[3]ikanovitadewi1@gmail.com

Abstrak: Telekomunikasi adalah salah satu industri, di mana pelanggan memerlukan perhatian khusus, oleh karena itu, manajemen di sebuah perusahaan telekomunikasi ingin kehilangan pelanggan model prediksi untuk efisien memprediksi berpotensi kehilangan pelanggan. Jaringan syaraf adalah metode yang sering digunakan untuk memprediksi. Teknik yang paling populer dalam metode adalah saraf algoritma jaringan *backpropagation.* Namun algoritma *backpropagation* memiliki kelemahan pada kebutuhan untuk data pelatihan besar dan optimasi yang digunakan kurang efisien. *Particle Swarm Optimization* (PSO) adalah suatu algoritma optimasi yang dapat memecahkan yang efektif masalah pada algoritma *neural network* umumnya menggunakan algoritma *backpropagation*. Pengujian model dengan berbasis menggunakan *Backpropagation Particle Swarm Optimization* menggunakan data pelanggan hilang pada telekomunikasi. Model yang dihasilkan diuji untuk memperoleh akurasi dan nilai-nilai AUC dari masing-masing algoritma untuk mendapatkan tes menggunakan nilai yang diperoleh akurasi *Backpropagation* adalah 85.48% dan nilai AUC adalah 0.531. Sementara pengujian dengan menggunakan *Backpropagation* berbasis *Particle Swarm Optimization* dipilih atribut dan penyesuaian nilai parameter yang diperoleh 86.05% akurasi dan nilai AUC adalah 0,637. Dengan demikian dapat disimpulkan bahwa data pelanggan uji hilang dalam telekomunikasi menggunakan aplikasi *Particle Swarm Optimization Backpropagation* dan dalam pemilihan atribut diperoleh bahwa metode ini lebih akurat dalam prediksi pelanggan hilang telekomunikasi dibandingkan dengan *Backpropagation*, ditandai dengan peningkatan akurasi 00:57% dan nilai-nilai AUC dari 0.106, dengan nilai yang dimasukkan ke dalam akurasi klasifikasi cukup.
Kata Kunci: Telekomunikasi, *Neural Network, Backpropagation, Particle Swarm Optimization*.

Abstract: Telecommunications is one of the industries, in which customers require special attention, therefore, management in a telecom company wants to lose customers prediction models to efficiently predict the potential loss of customers. Neural networks are often used method to predict. The most popular technique in the method is a neural network algorithm backpropagation. However backpropagation algorithm has a weakness on the need for a large training data and optimization is used less efficiently. Particle Swarm Optimization (PSO) is an optimization algorithm that can solve the problem of effective neural network algorithms are generally using the backpropagation algorithm. Model-based testing by using Particle Swarm Optimization Backpropagation using customer data lost on telecommunications. The resulting models were tested to obtain accuracy and AUC values of each algorithm to obtain test the accuracy of the values obtained using Backpropagation is 85.48 % and the AUC value was 0.531. While testing using Particle Swarm Optimization based Backpropagation selected attributes and adjusting parameter values obtained 86.05% accuracy and AUC value was 0.637. It can be concluded that the customer data lost in the telecommunications test applications using Particle Swarm Optimization Backpropagation applications and in the selection of attributes is obtained that this method is more accurate in predicting the missing customer telecommunications compared with Backpropagation, characterized by an increase in accuracy of 00:57% and AUC values of 0.106,

with a value that is inserted into the classification accuracy is quite.
Keywords: Telecommunications, Neural Network, Backpropagation, Particle Swarm Optimization.

I. PENDAHULUAN

Selama dekade terakhir, jumlah pengguna ponsel telah meningkat secara dramatis. Pada akhir tahun 2009 jumlah pengguna ponsel di seluruh dunia telah melampaui empat miliar, yang merupakan lebih dari 60% dari populasi dunia [1]. Telekomumikasi merupakan salah satu industri, dimana pelanggan sangat membutuhkan perhatian khusus karena sangat berpengaruh dalam mempertahankan kestabilan pendapatan, industri telekomunikasi selalu menghadapi ancaman kerugian finansial dari pelanggan yang hilang, oleh karena itu, manajemen di sebuah perusahaan telekomunikasi menginginkan model prediksi pelanggan hilang yang efisien untuk memprediksi pelanggan yang berpotensi hilang [2].

Dalam beberapa tahun terakhir ini telah terjadi banyak perubahan di industri telekomunikasi seperti adanya pasar bebas yang membuat persaingan yang sangat ketat dan para penyedia layanan telekomunikasi telah mengeluarkan layanan serta produk baru yang menyebabkan banyaknya *customer* telekomunikasi yang perpindah sehingga membuat penyedia layanan telekomunikasi merugi [3].

Prediksi pelanggan telekomunikasi yang hilang sudah pernah dilakukan, B. Huang, M. T. Kechadi, and B. Buckley melakukan prediksi pelanggan hilang dengan 7 (tujuh) metode prediksi (*Logistic Regressions, Linear Classifications, Naive Bayes, Decision Trees, MultilayerPerceptron Neural Networks, Support Vector Machines and the Evolutionary Data Mining Algorithm*) [3], W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens menggunakan 21 algoritma klasifikasi dan 11 dataset [1], B. Huang, B. Buckley, and T.-M. Kechadmelakukan prediksi pelanggan hilang berdasarkan pendekatan optimasi NSGA-II [4].

Decision tree mempunyai kelebihan yaitu mempunyai kelebihan dalam prediksi karena struktur algoritmanya mudah dimengerti dan tingkat kesalahannya cukup kecil sedangkan kelemahan algoritma *decision tree* adalah keandalan cabang yang lebih rendah menjadi lebih buruk dari cabang di atasnya, pohon keputusan yang dihasilkan tidak optimal dan tidak bisa menggunakan sampel yang lebih besar [5], karena itu tidak mudah untuk memahami pohon keputusan besar dan masalah *overfitting* data bisa terjadi dengan target data terbatas yang ditetapkan.

Neural network memiliki kelebihan pada prediksi non linear, memiliki *performance* yang sangat baik di parallel processing dan kemampuan untuk mentoleransi kesalahan [6]. Hal ini sangat tepat untuk karakteristik data prediksi pelanggan yang hilang pada penelitian ini. *Neural network* merupakan metode yang sering digunakan untuk memprediksi [7][8]. Teknik paling populer pada metode *neural network* adalah algoritma *backpropation* yang banyak digunakan untuk memecahkan banyak masalah di dunia nyata dengan membangun model terlatih yang menunjukkan kinerja yang baik dalam beberapa masalah non-linier [9]. Namun algoritma *backpropagation* mempunyai kelemahan pada perlunya data training yang besar dan optimasi yang digunakan kurang efisien [6]. Hal ini dapat dipecahkan karena jumlah data training pada penelitian ini sebanyak 3334 *record*.

Particle Swarm Optimization (PSO) merupakan algoritma optimasi yang efektif yang dapat memecahkan masalah yang ada pada algoritma *neural network* yang pada umumnya menggunakan algoritma *backpropagation* [9]. PSO memiliki

perbandingan lebih untuk pemilihan fitur dan memiliki kinerja lebih unggul untuk banyak masalah optimasi dengan lebih cepat dan tingkat konvergensi yang lebih stabil [10]. Karakteristik PSO adalah interaksi sosial yang mempromosikan pembagian informasi antara partikel yang akan membantu dalam pencarian solusi yang optimal [9]. PSO memiliki beberapa parameter seperti posisi, kecepatan, kecepatan maksimum, percepatan konstanta dan berat inersia. Dalam teknik PSO terdapat beberapa cara untuk melakukan pengoptimasian diantaranya: meningkatkan bobot atribut (*attribute weight*) terhadap semua atribut atau variabel yang dipakai, menseleksi atribut (*attribute selection*), dan *feature*.

Pada penelitian ini penulis mengusulkan penerapan PSO untuk memecahkan masalah yang terjadi pada *neural network* dengan memilih fitur pada bobot *atribut* untuk memaksimalkan kinerja dari model yang dihasilkan sehingga hasil prediksi pelanggan telekomunikasi yang hilang akan lebih akurat.

II. LANDASAN TEORI

A. *Neural Network*

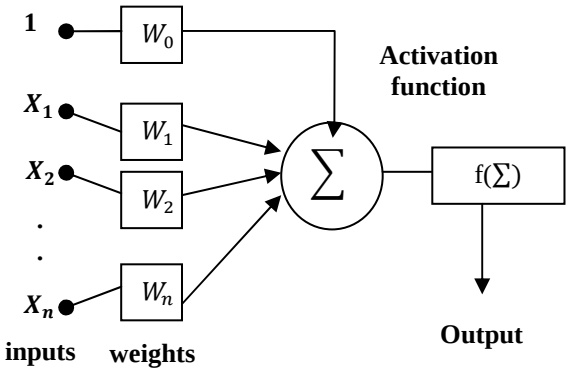
Neural Network merupakan *processor* yang terdistribusi paralel, terbuat dari unit-unit yang sederhana, dan memiliki kemampuan untuk menyimpan pengetahuan yang diperoleh secara eksperimental dan siap pakai untuk berbagai tujuan (S.Haykin, 1999) dalam [8]. *Neural network* ini meniru otak manusia dari sudut:

- Pengetahuan diperoleh oleh *network* dari lingkungan, melalui suatu proses pembelajaran.
- Kekuatan koneksi antar unit yang disebut *synaptic weights*, berfungsi untuk menyimpan pengetahuan yang telah diperoleh oleh *network* tersebut.

Pada tahun 1943, Mc. Culloch dan Pitts memperkenalkan model matematika yang merupakan penyederhanaan dari struktur sel saraf yang sebenarnya

$$y = f(\sum_{i=1}^{n} x_i W_i) \qquad (1)$$

Korelasi antara ketiga komponen pada persamaan di atas yaitu: Signal x berupa vektor berdimensi n (x1 , x2 ,…,xn)T akan mengalami penguatan oleh *synapse* w (w1, w2 ,…,wn)T. Selanjutnya akumulasi dari penguatan tersebut akan mengalami transformasi oleh fungsi aktifasi f. Fungsi f ini akan memonitor, bila akumulasi penguatan signal itu telah melebihi batas tertentu, maka sel *neuron* yang semula berada dalam kondisi "0", akan mengeluarkan signal "1". Berdasarkan nilai *output* tersebut (y), sebuah *neuron* dapat berada dalam dua status: "0" atau "1". *Neuron* disebut dalam kondisi *firing* bila menghasilkan output bernilai "1".



Gambar 1. *Mc. Culloch and Pitts Neuron Model*[5]

Gambar di atas memperlihatkan bahwa sebuah *neuron* memiliki tiga komponen:

1. *synapse* (w1 , w2 ,…,wn) T
2. alat penambah (*adder*)
3. fungsi aktifasi (f)

Sebuah *neural network* dapat dianalisa dari dua sisi:

1. Bagaimana *neuron-neuron* tersebut dirangkaikan dalam suatu jaringan (arsitektur)
2. Bagaimana jaringan tersebut dilatih agar memberikan *output* sesuai dengan yang dikehendaki (algoritma pembelajaran). Algoritma pembelajaran ini menentukan cara bagaimana nilai penguatan yang optimal diperoleh secara otomatis.

Berdasarkan arsitekturnya, *neural network* dapat dikategorikan, antara lain, *singlelayerneural network*, *multilayer neural network*, dan *recurrent neural network*. Berbagai algoritma pembelajaran antara lain *Hebb's law*, *Delta rule*, *Backpropagation algorithm*, dan *Self Organizing Feature Map*. Berawal dari diperkenalkannya model matematika *neuron* oleh Mc. Culloch dan Pitts, penelitian di bidang *neural network* berkembang cukup pesat, dan mencapai puncak keemasan pertama pada era tahun 60, dan puncak kedua pada pertengahan tahun 80-an.

Penelitian dalam bidang ini, dapat dibagi dalam tiga kategori:

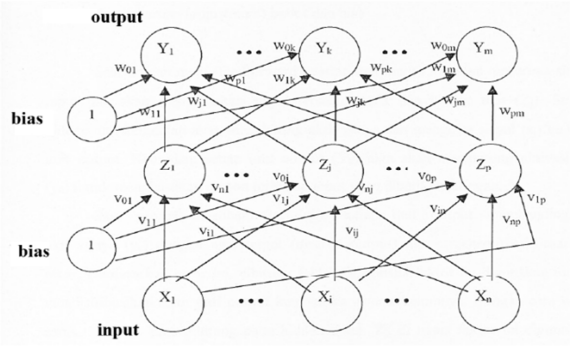
1. Riset untuk meneliti proses informasi yang terjadi pada otak dan jaringan saraf. Tema ini merupakan porsi penelitian para ahli medis dan *neuro scientist*.
2. Penelitian teoritis untuk mendalami konsep dasar proses informasi pada otak. Kategori ini memerlukan ketajaman analisa matematika untuk menggali dasar-dasar teori dari proses tersebut.
3. Penelitian yang bertujuan memanfaatkan teori-teori yang telah ada untuk aplikasi. Dalam hal ini, perlu sekali memperhatikan tingkat akurasi sistem, dan menekan biaya serendah mungkin *(low cost solution)*.

Dewasa ini, *neural network* telah diaplikasikan di berbagai bidang. Hal ini dikarenakan *neural network* memiliki kelebihan-kelebihan sebagai berikut:

1. Dapat memecahkan problema *non-linear* yang umum dijumpai di aplikasi
2. Kemampuan memberikan jawaban terhadap *pattern* yang belum pernah dipelajari *(generalization)*
3. Dapat secara otomatis mempelajari data numerik yang diajarkan pada jaringan tersebut.

B. Metode Backpropagation

Salah satu metode pelatihan terawasi pada neural network adalah metode *backpropagation*, di mana ciri dari metode ini adalah meminimalkan error pada *output* yang dihasilkan oleh jaringan. Pada gambar di bawah ini, *unit input* dilambangkan dengan X, *hidden unit* dilambangkan dengan Z, dan *unit output* dilambangkan dengan Y. Bobot antara X dan Z dilambangkan dengan v sedangkan bobot antara Z dan Y dilambangkan dengan w.



Gambar 2. Arsitektur Jaringan *Backpropagation*[8]

Penerapan *backpropagation network* terdiri dari 2 tahap:

1. Tahap pelatihan, di mana pada tahap ini diberikan sejumlah data pelatihan dan target
2. Tahap pengujian atau evaluasi, dilakukan setelah selesai tahap pelatihan

Pada intinya, pelatihan dengan metode *backpropagation* terdiri dari tiga langkah,yaitu:

1. Data dimasukkan ke *input* jaringan (*feedforward*)
2. Perhitungan dan propagasi balik dari *error* yang bersangkutan
3. Pembaharuan (*adjustment*) bobot dan bias.

Saat umpan maju (*feedforward*), setiap unit *input* (Xi) akan menerima sinyal *input* dan akan menyebarkan sinyal tersebut pada tiap *hidden unit* (Zj), setiap *hidden unit* kemudian akan menghitung aktivasinya dan mengirim sinyal (zj) ke tiap unit *output*. Kemudian setiap unit *output* (Yk) juga akan menghitung aktivasinya (yk) untuk menghasilkan respons terhadap *input* yang diberikan jaringan. Saat proses pelatihan (*training*), setiap unit *output* membandingkan aktivasinya (yk) dengan nilai target (tk) untuk menentukan besarnya *error*. Berdasarkan *error* ini dihitung faktor δ_k, di mana faktor ini digunakan untuk mendistribusikan *error* dari *output* ke *layer* sebelumnya. Dengan cara yang sama, faktor δ_j juga dihitung pada *hidden* unit Zj, di mana faktor ini digunakan untuk memperbaharui bobot antara *hidden layer* dan *input layer*. Setelah semua faktor δ ditentukan, bobot untuk semua *layer* diperbaharui. Notasi yang digunakan dalam algoritma pelatihan:

x = Data *training input* x = (x1,…,xi,…,xn)

t = Data *training* untuk *target output* t = (t1,…,tk,…,tm)

α = *Learning rate* yaitu parameter untuk mengontrol perubahan bobot selama pelatihan.

Xi = Unit *input* ke-i

Zj = *Hidden* unit ke-j

Yk = Unit *output* ke-k

v0j = Bias untuk *hidden* unit ke-j

vij = Bobot antara unit *input* ke-i dengan *hidden* unit ke-j

w0k = Bias untuk unit *output* ke-k

Wjk = Bobot antara *hidden* unit ke-j dengan unit *output* ke-k

δk = Faktor koreksi *error* untuk bobot wjk

δj = Faktor koreksi *error* untuk bobot vij

m = Momentum

Tahap-tahap pelatihan:

1. Tahap 0: Inisialisasi bobot dan bias. Baik bobot maupun bias dapat diset dengansembarang angka (acak) dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif).
2. Tahap 1: Jika *stopping condition* masih belum terpenuhi, jalankan tahap 2-9
3. Tahap 2: Untuk setiap data *training*, lakukan tahap 3-8
4. Tahap 3: Setiap unit *input* (Xi,i=1,…,n) menerima sinyal *input* xi dan menyebarkan sinyal tersebut pada seluruh unit pada *hidden layer*. Perlu diketahui bahwa *input* xi yang dipakai di sini adalah *input* training data yang sudah diskalakan.
5. Tahap 4: Setiap *hidden unit* (Zj,j=1,…,p) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya

$$z_{-in_j} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad (2)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan,

$$Z_j = f(z_{in_j}) \quad (3)$$

lalu mengirim sinyal *output* ini ke seluruh *unit* pada *unit output*

6. Tahap 5: Setiap *unit output* (Yk,k=1,…,m) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasnya,

$$y_{in_k} = W_{0k} + \sum_{j=1}^{p} Z_j W_{jk} \quad (4)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *unit output* yang bersangkutan

$$y_k = f(y_{in_k}) \quad (5)$$

7. Tahap 6: Propagasi balik *error* (*backpropagation of error*). Setiap *unit output* (Yk,k=1,…,m) menerima suatu *target* (*output* yang diharapkan) yang akan dibandingkan dengan *output* yang dihasilkan.

   $$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (6)$$

   Faktor δk ini digunakan untuk menghitung koreksi *error* (Δwjk) yang nantinya akan dipakai untuk memperbaharui wjk, di mana:

   $$\Lambda W_{jk} = \alpha\ \delta_k Z_j \quad (7)$$

   Selain itu juga dihitung koreksi bias Δw0k yang nantinya akan dipakai untuk memperbaharui w0k, di mana:

   $$\Lambda W_{0k} = \alpha\ \delta_k \quad (8)$$

   Faktor δk ini kemudian dikirimkan ke *layer* di depannya.

8. Tahap 7: Setiap *hidden unit* (Zj,j=1,…,p) menjumlah *input delta* (yang dikirim dari *layer* pada tahap 6) yang sudah berbobot.

   $$\delta\_in_j = \sum_{k=1}^{m} \delta_k W_{jk} \quad (9)$$

   Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi *error* δj, di mana:

   $$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (10)$$

   Faktor δj ini digunakan untuk menghitung koreksi *error* (Δvij) yang nantinya akan dipakai untuk memperbaharui vij, di mana:

   $$\Lambda V_{ij} = \alpha\ \delta_j x_i \quad (11)$$

   Selain itu juga dihitung koreksi bias Δv0j yang nantinya akan dipakai untuk memperbaharui v0j, di mana:

   $$\Lambda V_{0j} = \alpha\ \delta_j \quad (12)$$

9. Tahap 8: Pembaharuan bobot dan bias: Setiap *unit output* (Yk,k=1,…,m) akan memperbaharui bias dan bobotnya dengan setiap *hidden unit*.

   $$W_{jk}(baru) = W_{jk}(lama) + \Lambda W_{jk} \quad (13)$$

   Demikian pula untuk setiap *hidden unit* akan memperbaharui bias dan bobotnya dengan setiap *unit input.*

   $$V_{ij}(baru) = V_{ij}(lama) + \Lambda V_{ij} \quad (14)$$

10. Tahap 9: Memeriksa *stopping condition* Jika *stop condition* telah terpenuhi, maka pelatihan jaringan dapat dihentikan. Untuk menentukan *stopping condition* terdapat dua cara yang biasa dipakai, yaitu:
    1. Membatasi iterasi yang ingin dilakukan.
       Misalnya jaringan akan dilatih sampai iterasi yang ke-500. Yang dimaksud dengan satu iterasi adalah perulangan tahap 3 sampai tahap 8 untuk semua *training data* yang ada.
    2. Membatasi *error*.
       Misalnya menentukan besar *Mean Square Error* antara *output* yang dikehendaki dan*output* yang dihasilkan oleh jaringan. Jika terdapat sebanyak *m training* data, maka untuk menghitung *Mean Square Error* digunakan persamaan berikut:

$$MSE = 0{,}5\ x\ \{(tk1 - yk1)2 + (tk2 - yk2)2 + \cdots + (tkm - ykm)2\} \quad (15)$$

C. Particle Swarm Optimization (PSO)

PSO adalah teknik optimasi berbasis populasi yang dikembangkan oleh Eberhart dan Kennedy pada tahun 1995, yang terinspirasi oleh perilaku sosial kawanan burung atau ikan [9]. PSO dapat diasumsikan sebagai kelompok burung secara mencari makanan disuatu daerah. Burung tersebut tidak tahu dimana makanan tersebut berada, tapi mereka tahu sebarapa jauh makanan itu berada, jadi strategi terbaik untuk menemukan makanan tersebut adalah dengan mengikuti burung yang terdekat dari makanan tersebut (Salappa, Doumpos, &Zopounidis, 2007) dalam [5]. PSO digunakan untuk memecahkan masalah optimasi.

Serupa dengan algoritma genetik (GA), PSO melakukan pencarian menggunakan populasi (*swarm*) dari individu (partikel) yang akan diperbaharui dari iterasi. PSO memiliki beberapa parameter seperti posisi, kecepatan, kecepatan maksimum, konstanta percepatan, dan berat inersia. PSO memiliki perbandingan lebih atau bahkan pencarian kinerja lebih unggul untuk banyak masalah optimasi dengan lebih cepat dan tingkat konvergensi yang lebih stabil [9].

Untuk menemukan solusi yang optimal, masing-masing partikel bergerak kearah posisi yang terbaik sebelumnya dan posisi terbaik secara global. Sebagai contoh, partikel ke-i dinyatakan sebagai: *xi = (xi1, xi2,....xid)* dalam ruang ddimensi. Posisi terbaik sebelumnya dari partikel ke-i disimpan dan dinyatakan sebagai *pbesti = (pbesti,1, pbesti,2,...pbesti,d).* Indeks partikel terbaik diantara semua partikel dalam kawanan group dinyatakan sebagai gbestd. Kecepatan partikel dinyatakan sebagai: *vi = (vi,1,vi,2,....vi,d).* Modifikasi kecepatan dan posisi partikel dapat dihitung menggunakan kecepatan saat ini dan jarak pbesti, gbestd seperti ditunjukan persamaan berikut:

$$vi,d = w * vi,d + c1 * R * (pbesti,d - xi,d) + c2 * R * (gbestd - xi,d) \quad (16)$$

$$xid = xi,d + vi,d \quad (17)$$

Dimana:

Vi, d = Kecepatan partikel ke-i pada iterasi ke-i
w = Faktor bobot inersia
c1, c2 = Konstanta akselerasi (*learning rate*)
R = Bilangan random (0-1)
Xi, d = Posisi saat ini dari partikel ke-i pada iterasi ke-i
pbesti = Posisi terbaik sebelumnya dari partikel ke-i
gbesti = Partikel terbaik diantara semua partikel dalam satu kelompokatau populasi
n = Jumlah partikel dalam kelompok
d = Dimensi

Persamaan (1) menghitung kecepatan baru untuk tiap partikel (solusi potensial) berdasarkan pada kecepatan sebelumnya (Vi,m), lokasi partikel dimana nilai *fitness* terbaik telah dicapai (pbest), dan lokasi populasi *global* (*gbest* untuk versi *global*, *lbest* untuk versi *local*) atau *local neighborhood* pada algoritma versi *local* dimana nilai fitness terbaik telah dicapai. Persamaan (2) memperbaharui posisi tiap partikel pada ruang solusi. Dua bilangan acak c1 dan c2 dibangkitkan sendiri. Penggunaan berat inersia w telah memberikan performa yang meningkat pada sejumlah aplikasi. Hasil dari perhitungan partikel yaitu kecepatan partikel diantara interval [0,1] [9].

III. METODOLOGI

A. *Desain Penelitian*

Penelitian ini menggunakan jenis penelitian eksperimen, dengan tahapan penelitian sebagai berikut:

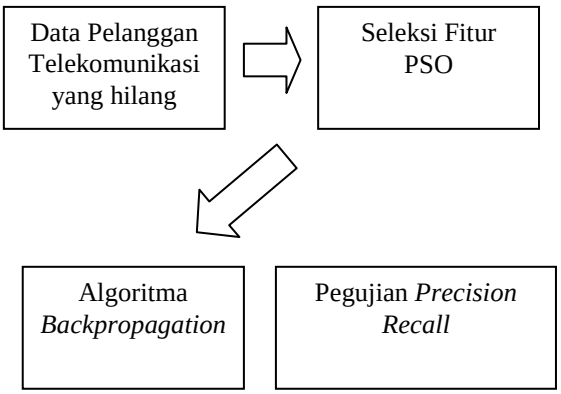
1. Pengumpulan Data
   Pengumpulan data merupakan langkah awal pada suatu penelitian. Data yang digunakan pada penelitian ini adalah dataset pelanggan telekomunikasi yang hilang.
2. Pengolahan Awal
   Pengolahan awal (*Preprocessing*) merupakan tahap untuk mempersiapkan data yang telah diperoleh dari tahap pengumpulan data, yang akan digunakan pada tahap selanjutnya.
3. Eksperimen dan Pengujian
   Tahapan ini akan membahas tahapan eksperimen dan teknik pengujian yang akan digunakan.
4. Evaluasi dan Validasi Penelitian
   Tahapan ini akan membahas hasil evaluasi dari eksperimen yang telah digunakan

B. Pengumpulan Data

Pada bagian ini di jelaskan data yang diperoleh memiliki 3334 *record* dan *field-field* yang ada pada data tersebut adalah: *State, Account Length, Area Code, Phone, Int Plan, VMail Plan, VMail Message, Day Mins, Day Calls, Day Charge, Eve Mins, Eve Calls, Eve Charge, Night Mins, Night Calls, Night*.

C. Desain Eksperimen

Model penelitian yaitu menerapkan algoritma *backpropagation* dan PSO yang akan membantu pada tahapan seleksi fitur. Data set dari pelanggan telekomunikasi yang hilang berasal dari file CSV diimport ke *Rapid Miner* untuk melakukan seleksi fitur meggunakan PSO dan diklasifikasi menggunakan Algoritma *Backpropagation* sehingga dapat menghasilkan hasil pegujian seperti yang terlihat pada gambar 3 berikut:



Gambar 3. Model ekperimen

D. Evaluasi Dan Validasi Hasil

Pada penelitian ini, penerapan algoritma *Backpropagation* dengan menentukan nilai *training cycle*, *learning rate*, dan momentum terlebih dahulu. Setelah didapatkan nilai akurasi dan AUC terbesar, nilai yang paling tinggi dari t*raining cycle*, *learning rate* dan momentum selanjutnya adalah menentukan ukuran (*size*) pada *hidden layer* tersebut. Sedangkan penerapan algoritma *backpropagation* berbasis PSO berdasarkan pada nilai training *cycle, learning rate*, dan momentum pada algoritma tersebut untuk menemukan peningkatan akurasi serta nilai *precision* dan *recall* untuk evaluasinya.

IV. HASIL DAN PEMBAHASAN

A. Metode Neural Network (Backpropagation)

Pada penelitian ini nilai *training cycle*, momentum, dan *learning rate* ditentukan dengan cara melakukan uji coba memasukkan nilai dengan *range* 500,1000, dan 1500 untuk *training cycles*, serta nilai 0.1, 0.2, dan 0.3 untuk *learning rate* dan 0.4, 0.7, dan 0.9 untuk momentum.

1) Hasil Pengujian *Backpropagation: Confusion Matrix* Berdasarkan data *training* yang diolah sebanyak 3334 *record* diperoleh hasil akurasi 85,48%, *precision* 46,67%, *recall* 3,53%.
2) *Hasil pengujian Backpropagation Berbasis Particle Swarm Optimization: Confusion Matrix* Berdasarkan data yang diolah sebanyak 3334 *record* diperoleh hasil sebagai berikut: Nilai akurasi dengan *neural network* berbasis PSO Jumlah *True Positive* (tp) adalah 2773 *record* diklasifikasikan sebagai *No Churn* dan *False Negative* (fn) sebanyak 388 *record* diklasifikasikan sebagai *No Churn* tetapi *True Churn*.

   Berikutnya 95 *record* untuk *True Negati*ve (tn) diklasifikasikan sebagai *Churn*, dan 77 *record False Positive* (fp) diklasifikasin sebagai *Churn* ternyata *No Churn*.

Berdasarkan gambar menunjukan bahwa, tingkat akurasi dengan menggunakan algoritma *backpropogation* berbasis PSO adalah sebesar 86.05 %, dan dapat dihitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*, ppv, dan npv hasilnya pada persamaan dibawah ini:

$$acc = \frac{tp + tn}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \qquad acc = \frac{2773 + 95}{2773 + 95 + 77 + 388}$$

$$sensitifity = \frac{tp}{\text{tp} + \text{fn}} \qquad sensitifity = \frac{2773}{2773 + 388}$$

$$specificity = \frac{tn}{\text{tn} + \text{fp}} \qquad specificity = \frac{95}{95 + 77}$$

$$PPV = \frac{tp}{\text{tp} + \text{fp}} \qquad PPV = \frac{2773}{2773 + 77}$$

$$NPV = \frac{tn}{\text{tn} + \text{fn}} \qquad NPV = \frac{95}{95 + 388}$$

Hasil perhitungan dari persamaan diatas terlihat pada Tabel 1 dibawah ini:

Tabel 1. Hasil Perhitungan

Accuracy	86.05%
Sensitifity	87.72%
Specificity	55.23%
PPV	97.29%
NPV	19.66%

B. Pembahasan Analisa Hasil Eksperimen

Berdasarkan eksperimen yang dilakukan dapat disumpulkan bahwa PSO dapat mangatasi masalah optimasi pada algoritma *backpropagation* pada prediksi pelanggan yang hilang, dapat dilihat dari peningkatan akurasi yaitu dengan algoritma *backpropagation* menghasilkan akurasi sebesar 85.48% dan AUC sebesar 0.531, kemudian peningkatan terjadi pada saat di tambahkan metode optimasi PSO yaitu akurasi dan AUC meninggkat menjadi 86.05%dan 0.637. Peningkatan tersebut terjadi karena metode optimasi PSO melakukan pencarian solusi optimal sampai semua partikel memiliki skema solusi yang sama atau ketika iterasi maksimum sudah tercapai sehigga dapat meningkatkan nilai akurasi.

V. KESIMPULAN

Dalam penelitian ini dilakukan pengujian model dengan menggunakan *Backpropagation* berbasis *Particle Swarm Optimization* dengan menggunakan data pelanggan yang hilang pada telekomunikasi. Model yang dihasilkan diuji untuk mendapatkan nilai *accuracy* dan AUC dari setiap algoritma sehingga didapat pengujian dengan menggunakan *Backpropagation* didapat nilai *accuracy* adalah 85.48 % dan nilai AUC adalah 0.531. Sedangkan pengujian dengan mengunakan *Backpropagation* berbasis *Particle Swarm Optimization* dilakukan seleksi atribut dan penyesuaian pada parameter didapatkan nilai *accuracy* 86.05% dan nilai AUC adalah 0.637 Maka dapat disimpulkan pengujian data pelangagan yang hilang pada telekomunikasi menggunakan *Backpropagation* dan penerapan *Particle Swarm Optimization* dalam pemilihan atribut didapat bahwa metode tersebut lebih akurat dalam prediksi pelangagan yang hilang pada telekomunikasi dibandingkan dengan *Backpropagation*, ditandai dengan peningkatan nilai akurasi sebesar 0.57% dan nilai AUC sebesar 0.106.

VI. SARAN

Diharapkan pada penelitian berikutnya dapat dikembangkan dengan menggunakan metode klasifikasi yang lain seperti *Decision Tree, Super Vector Machine* (SVM), dan lainnya atau menggunakan metode optimasi lainnya seperti *Genetic Algorithm* (GA).

REFERENSI

[1] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach," *European Journal of Operational Research*, vol. 218, no. 1, pp. 211–229, Apr. 2012.

[2] A. Idris, M. Rizwan, and A. Khan, "Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1808–1819, Nov. 2012.

[3] B. Huang, M. T. Kechadi, and B. Buckley, "Customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1414–1425, Jan. 2012.

[4] B. Huang, B. Buckley, and T.-M. Kechadi, "Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3638–3646, May 2010.
[5] M. Badrul, "Prediksi Hasil Pemilu Legislatif Dki Jakarta Dengan Metode Neural Network Berbasis Particle Swarm Optimization," 2012.
[6] X. Shao, "Based on Two Swarm Optimized Algorithm of Neural Network to Prediction the Switch's Traffic of Coal," *2011 International Symposium on Computer Science and Society*, pp. 299–302, Jul. 2011.
[7] C.-F. Tsai and Y.-H. Lu, "Customer churn prediction by hybrid neural networks," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12547–12553, Dec. 2009.
[8] Y. Fitrianto, "Penerapan neural network untuk prediksi data time-series arus lalu lintas jangka pendek," 2011.
[9] T.-S. Park, J.-H. Lee, and B. Choi, "Optimization for Artificial Neural Network with Adaptive inertial weight of particle swarm optimization," *2009 8th IEEE International Conference on Cognitive Informatics*, pp. 481–485, Jun. 2009.
[10] S. H. Ling, H. T. Nguyen, and K. Y. Chan, "A New Particle Swarm Optimization Algorithm for Neural Network Optimization," *2009 Third International Conference on Network and System Security*, pp. 516–521, 2009.